Synthesizing Sounds with Specified, Time-Varying Spectra *

Miller Puckette
Department of Music and Center for Research in Computing and the Arts, UCSD

Abstract

A unified framework is developed in which to compare several techniques for synthesizing sounds with desired spectra, using AM, FM, waveshaping, and pulse width modulation.

1 Introduction

Of the many approaches to specifying and synthesizing musical sounds, one of the oldest and best is to specify the sound's partial frequencies and spectral envelope. The frequencies of the partials might be chosen to lie on the harmonics of a desired fundamental frequency, and this gives a way of controlling the sound's (possibly time-varying) pitch. The spectral envelope is used to determine the amplitude of the individual partials, as a function of their frequencies, and is thought of as controlling the sound's (possibly time-varying) timbre. A simple example of this is synthesizing a plucked string as a sound with harmonically spaced partials in which the spectral envelope starts out rich but then dies away exponentially with higher frequencies decaying faster than lower ones, so that the timbre mellows over time. In a similar vein, [Risset] and [Grey] proposed spectral-evolution models for various acoustic instruments. A more complicated example is the spoken or sung voice, in which vowels appear as spectral envelopes, dipthongs and many consonants appear as time variations in the spectral envelopes, and other consonants appear as spectrally shaped noise.

Partly because of the intrinsic interest of the human voice and partly because of Bell Laboratories's strong influence on the early development of computer music, synthetic vocal sounds are perennial features of both early and modern computer music repertory. The palette of synthesis techniques has offered much variety within the framework described above. Starting with Mathews's *Daisy*, and soon afterward in Dodge's *Speech Songs*, subtractive synthesis has been used. At first the filters were usually designed using analyses of recorded voices (first via vocoder as in these two examples, and later using LPC as in Lansky's *Six Fantasies on a Poem by Thomas Campion*).

The other main historical approach to vocal synthesis (and also to synthesis of other time-varying spectra) has been the direct computation of formants, or more exactly, sounds containing a single formant that could be combined to create multi-formant spectra. In this class fall Konig's VOSIM generator [Templaars], Bennett and Rodet's FOF [Rodet] and Chowning's synthesis of formants using FM. A representative musical example of FOF synthesis is Barriere's *Chreode I*, and of Chowning's technique, his own piece, *Phoné*.

In these direct synthesis techniques, analyzed time-varying spectral envelopes have mostly given way to what Bennett and Rodet call "synthesis by rule," in which formantic placement is codified as a function of desired phonemes. (For that matter, synthesis by rule has also been applied to subtractive synthesis. On the other hand, analyses of continuous speech have not often been used to drive formant generators.)

Especially in *Phoné*, the listener is struck by a much greater sophistication of timbral control offered by the rule-based approach to speech synthesis. In addition, the direct synthesis of formants has, at least in the past, reached higher levels of sound quality

*Reprinted from *Proceedings*, ICMC 2001.

than has subtractive synthesis, which tends to sound machine-like and "buzzy," especially when compared to the FM approach.

In light of this, it is important to point out an important practical difficulty in Chowning's method, which is that there is no obvious way to cause the formants to slide upward or downward in frequency as they do in real speech and singing. This limitation, and some techniques for overcoming it, are considered in the sections that follow.

2 Carrier/modulator model

The FM algorithm is to calculate time-dependent values of the function,

$$x(t) = \cos(\omega_2 t + r\cos(\omega_1 t))$$

where ω_2 is the carrier frequency (in appropriate units), ω_1 is the modulating frequency, and r is the index of modulation. This formula only holds when the frequencies are not time-varying, but we can use it to derive or specify steady-state spectra which will still appear in the time-varying case.

To analyse the resulting spectrum we can write,

$$x(t) = \cos(\omega_2 t) * \cos(r\cos(\omega_1 t)) +$$

$$+(\text{another similar term}),$$

so, following [Lebrun] we can consider it as a sum of two waveshaping generators each essentially of the form,

$$m(t) = \cos(r\cos(\omega_1 t))$$

each ring modulated by a "carrier" signal of the form,

$$c(t) = \cos(\omega_2 t).$$

In Chowning's scheme for synthesizing formants, ω_2 becomes the formant center frequency and the modulator $m(t)$, which is aliased to a point in the spectrum centered at ω_2, determines the formant bandwidth and the placement of partial frequencies around the formant frequency. In particular, to get harmonically spaced partials we set the modulating frequency ω_1 to the fundamental and the carrier frequency to the multiple of ω_1 closest to the desired center frequency.

The bandwidth can then be controlled by changing the index of modulation. The center frequency ω_2 can't be changed continuously however, since for harmonicity it must be an integer multiple of ω_1.

The next section will describe two alternative forms of the carrier signal $c(t)$, each of which allow changing center frequency without losing harmonicity. In the following section we will consider alternative forms of the modulator $m(t)$ which in some situations are preferable to the classic FM formulation.

3 The carrier signal

Two workable strategies for producing "glissable" carrier signals have emerged, one simple, the other more complex but better at handling the case of very small bandwidths. In both cases, we start by synthesizing a low-bandwidth carrier signal with components clustered around the desired formant center frequency. The spectrum can then be fattened to a desired bandwidth by using a suitable modulator. We will now let ω_2 denote a desired center frequency, no longer necessarily an integer multiple of the fundamental frequency ω_1. We will denote the desired waveform period by

$$\tau = \frac{2\pi}{\omega_1}.$$

The first technique is to let

$$c(t) = w(t - \mathrm{ROUND}(t))\cos(\omega_2(t - \mathrm{ROUND}(t)))$$

where $\mathrm{ROUND}(t)$ denotes the nearest multiple of τ to t, and $w(t)$ is a windowing function such as the Hanning window:

$$w(t) = \frac{1}{2}(1 + \cos(\frac{2\pi t}{\tau})), -\frac{\tau}{2} \le t \le \frac{\tau}{2}.$$

In words, the signal $c(t)$ is simply a sample of a cosine wave at the desired center frequency, repeated at the (unrelated in general) desired period, and windowed to take out the discontinuities at period boundaries.

Here the full 6-DB bandwidth of the signal $c(t)$ will be $2 * \omega_1$, which is reasonably small but not negligible. (It is tempting to try to reduce bandwidth further by lengthening the "samples" and overlapping them, but this leads to seemingly insoluble phase cancellation problems.)

This method leads to an interesting generalization, which is to take a sequence of samples of length τ, align all their component phases to those of cosines, and use them in place of the cosine function in the formula for $c(t)$ above. The phase alignment is necessary to allow coherent cross-fading between samples so that the spectral envelope can change smoothly. If, for example, we use successive snippets of a vocal sample as input, we get a strikingly effective vocoder.

The second technique, first described in [Puckette], is to synthesize a carrier signal,

$$c(t) = a\cos(n\omega_1 t) + b\cos((n+1)\omega_1 t)$$

where $a + b = 1$ and n is an integer, all three chosen so that

$$(n + b) * \omega_1 = \omega_2,$$

so that the spectral center of mass of the two cosines is placed at ω_2. (Note that we make the amplitudes of the two cosines add to one instead of setting the total power to one; we do this because the modulator will operate phase-coherently on them.)

However, it is not appropriate simply to change a, b, and n as smooth control signals. The trick is to note that $c(t) = 1$ whenever t is a multiple of τ, regardless of the choice of a, b, and n as long as $a+b = 1$. Hence, we may make discontinuous changes in a, b, and n once each period without causing discontinuities in $c(t)$.

In the specific case of FM, if we wish we can now go back and modify the original formulation:

$$a\cos(n\omega_2 t + r\cos(\omega_1 t)) +$$

$$+ b\cos((n+1)\omega_2 t + r\cos(\omega_1 t)).$$

This is how to add glissandi to Chowning's original FM voices.

4 The modulator

In the waveshaping formulation the shape of the formantic peak is determined by the modulator term $m(t)$. In the case of FM this gives the famous Bessel "J" functions. At indices of modulation less than about 1.43 radians we get a proper bell-shaped spectrum, with bandwidths ranging from 0 to about $4\omega_1$ (full width at -6 dB height.) Further increases in index give rise to the well-known sidelobes in the FM spectrum.

Although we might desire the sidelobe effect, we needn't be tied to it; other possibilities abound. The formula for the general modulation signal is:

$$m(t) = \cos(r * F(\cos(\omega_1 t))).$$

We have so far found two functions:

$$F_1(x) = 1/(1 + x^2)$$

and

$$F_2(x) = \exp(-x^2)$$

which give rise to bell shaped formants at any index, without any sidelobes and also producing phase-coherent partials without changes in sign of the amplitudes of the components. In the case of F_1 the resulting spectra are particularly simple to describe; the component amplitudes drop off linearly in dB with distance from the center frequency. F_2 gives rise to "I" Bessel functions, which unlike FM's "J" functions do not give rise to sidelobes, and whose tails drop off more quickly than for F_1.

Since both of these are even functions, we set ω_1 to be half of the fundamental frequency, unlike the FM case where we set ω_1 to the fundamental; this accounts for there being only one term to calculate here instead of the two in our analysis of FM.

Yet another approach is pulse width modulation, for instance:

$$m(t) = w(rt) + w(r \cdot (t - \tau)) + w(r \cdot (t - 2\tau)) + ...$$

where $w(t)$ is the Hanning window defined above. This gives in effect a train of Hanning window functions, whose duty cycle is $1/r$. If we don't wish the windows to overlap we require r to be at least 1, and so the full 6-dB bandwidth is limited below by twice the fundamental frequency. If desired, we can allow double overlap (by dedicating one oscillator to the odd-numbered pulses and a second to the even ones); then the minimum bandwidth effectively drops to zero.

The spectrum is simply the Fourier transform of the Hanning window, which is approximately band-limited (actually only good to about -34 dB), as compared to the waveshaping solutions which are non-band-limited. If we desire better stop-band rejection than -34 dB we can pass to Blackman-Harris windows; in this case we must allow overlap 3 before we can attain zero bandwidth.

5 Noise

Up to now we have only synthesized discrete spectra. It is also sometimes desirable to synthesize "noisy" sounds with desired spectral envelopes. One technique for doing this is described in [Puckette]. The idea is to multiply a discrete spectrum (perhaps computed in one of the ways described above) with a noise signal with bandwidth ω_1. Each sinusoid is then modulated into a narrow noise band, and the overlapping noise bands fill out a continuous noisy spectrum.

However, the fact that each sinusoid is modulated by the *same* noise is problematic. To fix this we modulate four copies of the original signal, delayed varying amounts up to 10 milliseconds, by four independent band-limited noise streams. Each partial thus gets a different linear combination of the four noise signals and thus the partials "move" independently.

6 Implementation

These techniques have been gradually refined over the last fourteen years, using IRCAM's 4X and ISPW, and later the standard real-time interactive graphical synthesis environments Max/MSP, jMax, and Pd. The community of active users of the techniques has, however, remained quite small, at least partly since nothing has so far been published in computer music venues about it. Implementations in the form of external objects for Pd and Max/MSP are available, with sources, from `http://crca.ucsd.edu/~msp` and `http://crca.ucsd.edu/~tapel`.

References

[Grey] Grey, J.M. and Moorer, J.A., 1977. "Perceptual evaluations of synthesized musical instrument tones," J. Acoust. Soc. Am. 62, 454-462.

[Lebrun] Lebrun, M., 1979. "Digital waveshaping synthesis," *Journal of the Audio Engineering Society* 27/4, pp. 250-266.

[Puckette] Puckette, M. 1995. "Formant-based audio synthesis using nonlinear distortion." JAES 43/1, pp. 40-47. Reprinted on `http://crca.ucsd.edu/~msp`.

[Risset] Risset, J.C and Mathews, M.V., 1969. "Analysis of musical instrument tones," Physics Today 22, 23-40.

[Rodet] Rodet, X., Potard, Y., and Barriere, J.-B., 1984. "The CHANT project: from the synthesis of the singing voice to synthesis in general." *Computer Music Journal* 8/3, pp. 15-31.

[Templaars] Templaars, S., 1977. "The VOSIM signal spectrum," *Interface* 6, pp. 81-96.